VAASAN AMMATTIKORKEAKOULU
VASA YRKESHÖGSKOLA
UNIVERSITY OF APPLIED SCIENCES

Mengdi Wu

INTELLIGENT LINUX-BASED ACCESS POINT

Technology and Communication

2014

FOREWORD

This is my graduation thesis in the Degree Programme in Information Technology, VAMK, University of Applied Sciences.

I would like to express my gratitude to my supervisor, Dr. Chao Gao. He gave me the inspiration for my thesis. No work without efforts. During my entire thesis working time, he gave me a lot of useful guidance with patience and kindness. Also he taught me how to deal and solve the problems alone. This is a life time lesson for me. Moreover, during my undergraduate study time, as a Principal Lecturer of Telecommunications, his serious attitude for academic is setting a good model for me.

Last I would like to thank my family. Without their support, I could not have finished my studies this easy. However, I would also thank my friends Lv Chunqiu, Song Shuo for their help with my thesis. Hope they all will have a bright future.

Wu Mengdi

Vaasa, Finland

28/04/2014

VAASAN AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES
Information Technology

ABSTRACT

Author	Mengdi Wu
Title	Intelligent Linux-Based Access Point
Year	2014
Language	English
Pages	33+ 2 Appendices
Name of Supervisor	Chao Gao

Nowadays IEEE802.11-base WLAN (Wireless LAN also known as Wi-Fi) has been used everywhere. It is free to setup a Wi-Fi network for everybody, but sometime there can be more than 10 Wi-Fi networks in one area. For most of the networks, they work on “default” IEEE802.11 channels which are channel 1, 6 and 11. The effect of using same channel will lead to a bad Internet performance. We can easily improve it by choosing other channels, but which channel will be less crowded? I will tell you how my intelligent Linux-base Access Point works. This work will improve your network performance and also your neighbor’s network performance.

A Linux-based laptop, which requires both Ethernet and WLAN interfaces, was used in the thesis. This laptop was able to act as an Ad-Hoc AP (Access Point) and share the Internet through wireless via LAN.

By using the C language and Linux Shell Script, it was possible to find the WLAN channel situations and setup the WLAN interface as a WAP (Wireless Access Point). A program written in C language was used to calculate the Power Sum Interference of all the channels. The channel with minimum PSI will be the best channel in my area. By using the ready-made utilities in Linux, it was possible to find out the WLAN channel situation and set the WLAN interface as a WAP. The working channel, SSID and encryption method were also chosen.

After all the setup and calculation were done, Iperf was used to demonstrate that the performance of the WAP (Wireless Access Point) is “greatly” improved by using this application.

Keywords WLAN Channels, Linux, Iperf

CONTENTS

LIST OF FIGURES AND TABLES

LIST OF ABBREVIATIONS

WLAN	Wireless LAN also known as Wi-Fi
AP	Access Point
LAN	Local Area Network
WAP	Wireless Access Point
PSI	Power Sum Interference
ESSID	Extended Service Set Identification
SSID	Service Set Identifier
RSSI	Received Signal Strength Indicator
FCC	Federal Communications Commission
P2P	Peer to Peer
DHCP	Dynamic Host Configuration Protocol
WEP	Wired Equivalent Privacy
NAT	Network Address Translation
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
MSS	Maximum Segment Size
MTU	Maximum Transmission Unit
STA	Station (computer networking)

LIST OF APPENDICES

1 INTRODUCTION

1.1 Purpose of Thesis

Today's world is surrounded by networks. Networks means do not refer only to Internet networks, but also to the information, relationships and so on. According to all these facts, basically we rely on a "bridge" that can connect all the facts together. What is that "bridge" made of? From my point of view, the answer is telecommunication technology. In our everyday life, we are using WLAN (Wireless Local Area Network) as a method to get access with all kinds of information that we need. It is necessary for our daily life.

We all know that the wireless network is more flexible than wired network, but it is not as stable as the wired network. In wireless networking, we always have to consider all kinds of condition interferences. Sometime we can easily suffer from really bad signal and heavy interference.

In this case we really need to find a way out. There are few ways to improve the performance of our home wireless access point. In my research, I will focus on the WLAN channel allocation problem. By selecting the least crowded channel in one area, we try to find out the network performance difference between a default channel and a selected channel.

This thesis work proves that improving the Internet performance is not hard at all. It may only require the changing of the wireless channel.

1.2 Overview Structure of Thesis

This thesis is arranged as follows: Chapter 2 discusses the WLAN channel allocation and related issues. Chapter 3 addresses channel choosing algorithm based on WiFi standard RSSI. Chapter 4 lists out all the implementation methods. Finally Chapter 5 concludes the thesis. The relations between the chapters can be seen in Figure 1.

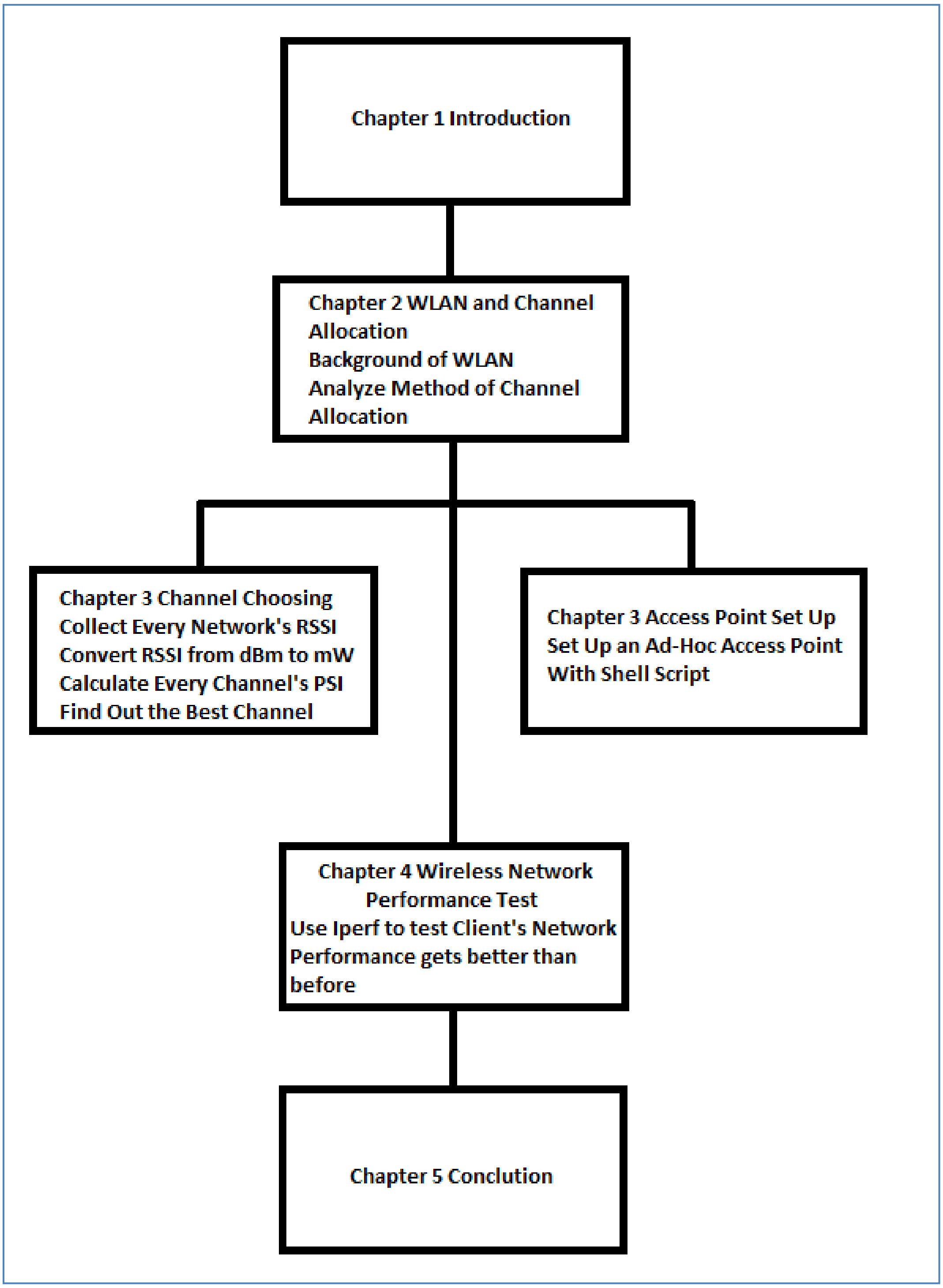


Figure 1. Overview Structure of Thesis

1.3 Methodology

Through the whole project the following tools and methods were used to achieve the goal:

1. A laptop with Linux Ubuntu 12.04 System. This laptop is working as WLAN AP.
2. A laptop with wireless card. It is working as WLAN client(s).
3. C Language. A program written in C language to calculate channel interference.
4. Linux Shell Script. A program to measure channel information and setup WLAN AP.
5. Iperf. This tool is used to compare the network performance under different settings.

2 WLAN AND ITS CHANNEL ALLOCATION

In this chapter, we mainly focus on the background of WLAN and the channel allocation problem. We will discuss WLAN structure, IEEE 802.11 standards, channels and interference.

In section 2.1 we talk about WLAN concepts and network structure. In section 2.2 we focus on the WLAN channel allocation method and analysis. This is the resolution to solve the WLAN interference.

2.1 Background

2.1.1 WLAN (Wireless Local Area Network)

WLAN is a wireless local area network which allows two or more mobile devices use Internet via the wireless connection. This connection is based on an access point which allows the users to move around within a certain coverage area. WLAN is also known as Wi-Fi, an IEEE802.11-base technology. It is easy to install and free to use. These attributes make Wi-Fi so popular in our life today/1/. The WLAN diagram shows in Figure 2.

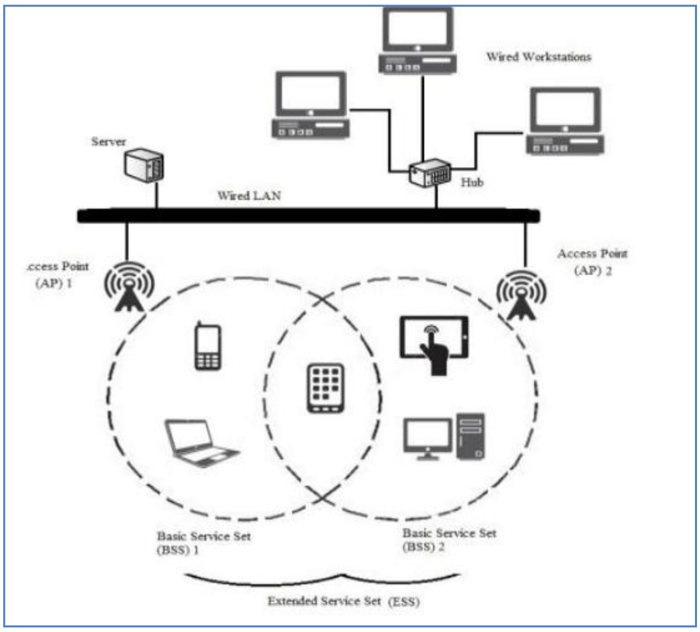


Figure 2. WLAN (Wi-Fi) Diagram/1/

2.1.2 IEEE 802.11 Standards

WLAN computer usually communicates in the 2.4, 3.6, 5 and 60 GHz frequency bands/1/. For 3.6 GHz frequency band, it is an extended operation of 802.11a. It is not mainly talked in this thesis. From Figure 3, we can see all the information that we may need to use. Today's computers are usually supported with IEEE 802.11b/g/n protocols.

802.11 Protocol	Release date	Frequency	Data rate per stream (Average)	Data rate per stream (Maximum)	Approximate indoor Range (m)	Approximate outdoor range (m)
Legacy	1997	2.4 GHz	1 Mbit/s	2 Mbit/s	20	100
802.11a	1999	5GHz	25 Mbit/s	54 Mbit/s	35	120
802.11b	1999	2.4 GHz	6.5 Mbit/s	11 Mbit/s	35	140
802.11g	2003	2.4GHz	25 Mbit/s	54 Mbit/s	38	140
802.11n	2009	2.4 GHz or 5 GHz bands	300 Mbit/s (20MHz*4 MIMO)	600 Mbit/s (40MHz*4 MIMO)	70	250

Figure 3. 802.11 Network RHY Standards

2.1.3 List of WLAN channels

Most of the IEEE 802.11 networks use 2.4 GHz frequency, because a longer wavelength radio has better range. As seen in Figure 4, there are 14 channels designated in the 2.4 GHz range spaced 5 MHz apart (with the exception of a 12 MHz spacing before Channel 14). /2/

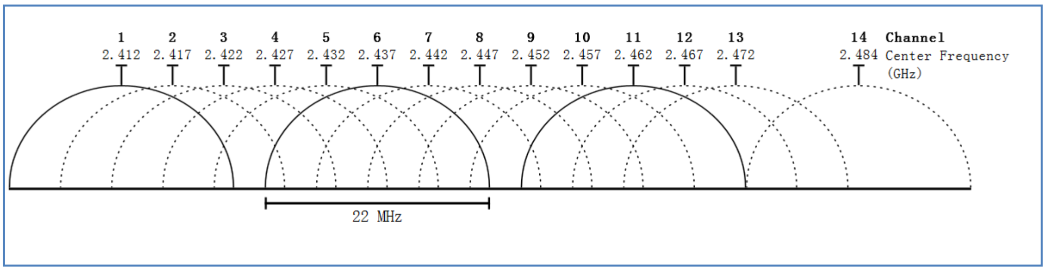


Figure 4. Graphical Representation of 2.4 GHz Band Channels Overlapping/3/

Due to the different requirements by regulatory authorities, not all the WLAN channels are in use in every country. In FCC or North American domain the 2.4-GHz band is divided down to 11 channels, but for the European or ETSI domain there are 13 channels. Figure 5 shows the Wi-Fi channels in use around the world.

Channel Number	Europe (ETSI)	North America (FCC)	Japan
1	✔	✔	✔
2	✔	✔	✔
3	✔	✔	✔
4	✔	✔	✔
5	✔	✔	✔
6	✔	✔	✔
7	✔	✔	✔
8	✔	✔	✔
9	✔	✔	✔
10	✔	✔	✔
11	✔	✔	✔
12	✔	No	✔
13	✔	No	✔
14	No	No	802.11b only

Figure 5. Wi-Fi Channels in Different Parts of the World

2.1.4 Ad Hoc (Peer to Peer)

The IEEE 802.11 has two modes, which are ad hoc and infrastructure mode. The Ad hoc network is a multi-hop, no infrastructure and self-organizing network, also known as peer to peer (P2P). Mobile units transmit directly peer to peer. It is not the same as a Wi-Fi Direct network. The Ad Hoc diagram can be seen in Figure 6. The entire network has no fixed infrastructure. Every node is movable and able to keep in touch with other nodes dynamically. Because of the limitation of the terminal radio coverage, the user's terminals cannot communicate with each other directly but they have to use other nodes for packet forwarding. Each node is also acting as a router; it is be able to maintain the function with other routing nodes.

Figure 6. Ad Hoc Diagram

Ad Hoc networks shows as a new network, we can set up a mobile communication network at any time, any place. It does not need a hardware-base to support the network infrastructure. It does not rely on the existing network communication facilities. All these features are very suitable to use when a disaster happens. The Ad Hoc network is mainly used for temporary communication needs. Compared to a wired network, its survival time is shorter.

Due to there being no wired infrastructure support in the Ad Hoc network, the communication between all the hosts are completed through wireless transmission. But the physical properties of a wireless channel provide the wireless transmission on a much lower bandwidth, compared to the cable channel. In addition, there is always interference in WLAN. The actual bandwidth of the mobile terminal should be much less than the theoretical maximum bandwidth.

2.1.5 Infrastructure

Infrastructure is a work mode with a wireless station (STA). In IEEE 802.11, a STA is present as a device which is capable to use the 802.11 protocol, such as a laptop, a desktop PC, AP or Wi-Fi phone. The infrastructure mode wireless network is working via an access point and regarding the AP as a traditional LAN hub function.

There are some differences between the infrastructure and the Ad Hoc mode. For infrastructure mode, it needs a fixed central control which Ad Hoc does not. Ad Hoc has a better self-organizing ability. When it is multi-routing, Ad Hoc needs a common routing node to achieve; but the infrastructure needs a dedicated router to complete. Also in topology, the infrastructure uses generally conventional topologies, such as heart-shaped, ring and so on. These are all static settings. Ad Hoc is totally opposite, it uses dynamic implementation.

2.1.6 WLAN Interference

It is generally thought that channel 1, 6 and 11 (in some countries channel 14) are non-overlapping channels. Obviously it is a misunderstanding. This kind of thought is too shallow. Actually 802.11b and 802.11g do not specify the bandwidth of every channel. They only declared the central frequency and spectral mask of every channel. The spectral mask defines the permitted power distribution across each channel. For 802.11b, it is shown in Figure 7 the spectral mask requires that the signal must drop 30 dB and 50 dB below its peak power when operating at±11 MHz and±22 MHz apart from the central frequency, respectively /4/.

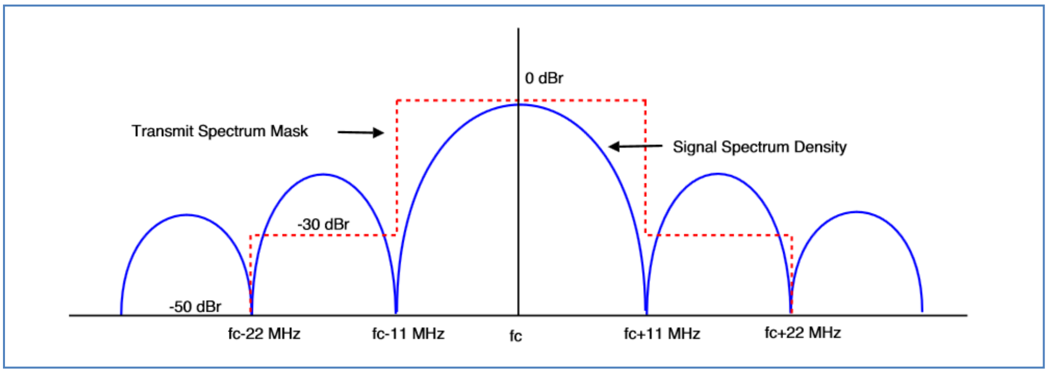


Figure 7. Power Spectrum Mask of the 2.4 GHz ISM /4/

The spectral mask only specifies the power limitation at $\pm$22 MHz, so is thought the bandwidth will not reach the limits. In a real life situation, when the distance between the transmitter and the receiver is too short, the effective power spectrum can be more than 22 MHz, so there is no non-overlapping channel. We have to say the channel 1, 6 and 11 effects each other less than other channels do.

We always have to consider the real life situation, such as how many users are in that channel and how strong signal that channel has. For wireless networks using the 2.4 GHz radio frequency, we should avoid the interferences which include the following: /5/

1. Microwave ovens.
2. Bluetooth enabled devices.
3. 2.4GHz wireless phones.
4. Other nearby wireless networks.
5. Electrical devices such as power lines.
6. Large metal cabinets (for instance fume cupboards in science labs)
7. Excessive metal pipe work.
8. Any other devices that operate in the 2.4 GHz bandwidth.

There are always some resolutions that can improve the interference. If there is a poor signal and strong interference, here are some tips to solve it /5/:

1. Change wireless channel.
2. Analyze the potential for interference.
3. Provide wider wireless coverage.
4. Deploy 802.11a.
5. Physical location (Put your wireless router high up and closes to the area you usually stay and use the Internet.)

2.2 Channel Allocation Method and Analyze

The Linux ready-made utilities help to find out the most important parameter Received Signal Strength Indicator (RSSI). It is the measurement of the power present in a received radio signal. /7/ In an IEEE 802.11 system, RSSI is the relative received signal strength in a wireless environment. When the RSSI is higher, it means the signal is much stronger. Although the IEEE 802.11 standard does not define any relationship between RSSI value and power level in mW or dBm, but by calculating the PSI an answer will be obtained.

As stated before, changing the wireless channel can be a solution. If we need to define which channel has the best performance, Power Sum Interference (PSI) can be calculated. For the PSI, we need to convert RSSI from dBm to mW and add up the entire network in one channel. The one with the minimum number is the best channel to choose.

$$RSSI = 1mW \times 10^{\frac{xdBm}{10}}$$

$$PSI_X = All\ the\ networks'\ RSSI(mW)\ in\ one\ channel$$

Usually one channel is overlapping with few channels, but in this situation we only take its adjacent channels. 802.11b and 802.11g do not specify the bandwidth of every channel. They only declare the central frequency and spectral mask of every channel. Which means when the user is using a channel, the user's RSSI is changing and the position of the user is moving all the time. It is hard to say how much they will affect each other when the channels are far away from one another.

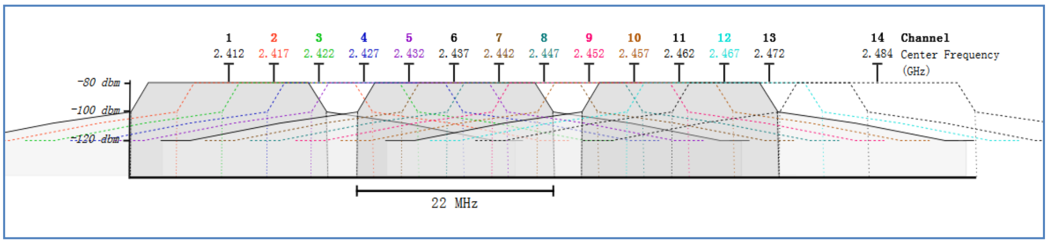


Figure 8. Spectral Masks for 802.11g in the 2.4 GHz Band/6/

For example, the channels can be seen in Figure 8, if we choose channel 1 to calculate. The energy overlap channel should be considered is channel 2. So we only need to collect all the RSSI in channel 1 and 2 to calculate the PSI in channel 1. Channel 3 is not included because channel 3 and channel 1 overlap less than half of the channel. When using channel 3, we do not know exactly where the user it is. It may not even in the range of channel 1. So we only consider the channel with its adjacent channels.

After calculating the PSI for all channels, it is easy to find out which channel is less crowded and has better performance. The lower PSI, the better quality the channel has.

Due to the manufacture setting, my Linux laptop only supports 11 channels; in this case we only tested with 11 channels. Figure 9 shows the power overlaps channels for each channel.

Channel	Center Frequency	Frequency delta	Channel Width (GHz)	Energy overlaps channels
1	2.412	5MHz	2.401–2.423	2
2	2.417	5MHz	2.406–2.428	1,3
3	2.422	5MHz	2.411–2.433	2,4
4	2.427	5MHz	2.416–2.438	3,5
5	2.432	5MHz	2.421–2.443	4,6
6	2.437	5MHz	2.426–2.448	5,7
7	2.442	5MHz	2.431–2.453	6,8
8	2.447	5MHz	2.436–2.458	7,9
9	2.452	5MHz	2.441–2.463	8,10
10	2.457	5MHz	2.446–2.468	9,11
11	2.462	5MHz	2.451–2.473	10

Figure 9. 802.11 b/g Channel to Frequency Map

2.3 Summary

With the method of channel allocation, we refer a certain calculation method to the implementation method. It helps to find out which channel is the least crowded in the area.

3 IMPLEMENTATION METHOD

3.1 Channel Selecting Calculation

3.1.1 Introduction

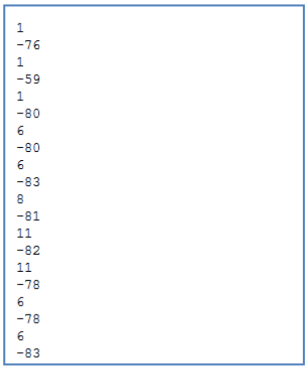
In this section, the best channel is found for the access point. Due to the WLAN interference, wireless networking is not as stable as wired Internet. When we suffer from the poor signal and strong interference, there are multiple ways to solve it. Here a wireless channel is chosen to fix the problem. The Linux ready-made utilities help to find out the other user's wireless connection situation. It is a good procedure to calculate the PSI for each channel and find out the best channel in this area.

3.1.2 Configuration

1. We use Linux command is used to collect all networks' RSSI. It will store in a file called "result.txt" in this case.

   ```
   sudo iwlist wlan0 scan|grep -o -e 'Channel:\([0-9]\+\)' -e 'Signal level=\([-0-9]\+\)
   ```

   With `iwlist wlan0 scan` command all the networks' RSSI and other information is collected, such as "address, channel's number, mode, encryption key statue and so" in the area, but the command "`grep -o -e 'Channel:\([0-9]\+\)' -e 'Signal level=\([-0-9]\+\)`" will only fetch the channel number and its signal level. The result shows in Figure 10.



```
1
-76
1
-59
1
-80
6
-80
6
-83
8
-81
11
-82
11
-78
6
-78
6
-83
```

Figure 10. "result.txt" outputs

2. Then the C language is used to get all the parameters that needed.
3. All the RSSI from dBm are converted to mW.

```
for(i=0;i<chlength;i++)
{
            PSI[i]=0.0;
            Power[i]=pow(10.0,RSSIpro[i]/10.0);
            printf("Channel  %d  ->  %.2fdBm->  %.20fmW\n",
charypro[i],RSSIpro[i],Power[i]);
```

This is a mathematical algorithm in the C language. Every piece of data is converted in result.txt from dBm to mW first. At this point every network's RSSI in mW is known.

4. PSI for every network is calculated.

```
double PSIsum[i];
      for(i=1;i<=11;i++)
      {
            PSIsum[i]=PSI[i-1]+PSI[i]+PSI[i+1];

            printf("Channel       %d->       PSI=%.20fmW\n",
i,PSIsum[i]);
      }
```

In these lines, every channel's RSSI in mW can be added up.

5. All the PSI are compared and the minimum one is looked for. The result is the best channel number in the area.

```
double x=PSIsum[0];
int best;
      for(i=1;i<=11;i++)
      {

    if (PSIsum[i]<x) {x=PSIsum[i];best=i;}

      }
      for(i=1;i<=11;i++)
      {
            if (PSIsum[i]==x)

      printf("Channel %d have better signal\n",i);
      }
```

3.1.3 Summary

In this section, the C programming is used to do all the calculation. Finally, the best channel number is printed out. Due to the Linux command result, every network's RSSI is received only in dBm, but the dBm cannot be multiplied together.

The power interference is calculated, so all the parameters should convert from dBm to mW. Also the power overlap channel for each channel should be also included when the power interference for each channel is calculated.

3.2 Ad Hoc Establishment

3.2.1 Introduction

In this section, a wireless access point is set up. By using both Ethernet and WLAN interfaces, it is possible to share the wired connection through the access point we established. Here the basic commands "`iwconfig`, `ifconfig`" are used to turn the Ad Hoc access point on. In addition, establishing the access point, it is also possible to customize the user name, the channel number and the encrypt password.

3.2.2 Configuration

WLAN configuration is done on an Ubuntu 12.04. The "`iwconfig`" command is used to configure the WLAN interface. Details of configuration are given in the following list.

1. Use `iwconfig` command to find out our WLAN interface's name. In this case the WLAN interface called wlan0. Then bring down the WLAN interface.
   ```
   sudo iwconfig
   sudo ifconfig wlan0 down
   ```
2. Switch the card into Ad Hoc mode.
   ```
   sudo iwconfig wlan0 mode ad-hoc
   ```
3. Customize the ESSID, channel number and add a WEP encryption key.
   ```
   sudo iwconfig wlan0 essid 'name'
   sudo iwconfig wlan0 channel 1
   sudo iwconfig wlan0 key 1234567890
   ```

4. Activate the interface and set the DHCP range.

   ```
   sudo ifconfig wlan0 up
   dnsmasq –dhcp-range= "192.168.7.101,192.168.7.110"
   ```

   Here DHCP server is added, this range must not clash with the LAN IP address. The range defines the capability of the AP that means how many clients can connect to it.

Now the client machine will be able to see and connect to the server, but there is no Internet connection through the wireless connection. By the later procedure, the wired Internet is enabled to go through the wireless. The DHCP range will limit the amount of the client machines.

5. Set up the iptables. All the old rules first must be removed. Then enable receiving and sending data packets and also the NAT on the server.

   ```
   iptables –F
   ```

   Remove all the rules one by one.

   ```
   iptables -P INPUT ACCEPT
   ```

   Enable to receive data packets for the host.

   ```
   iptables -P FORWARD ACCEPT
   ```

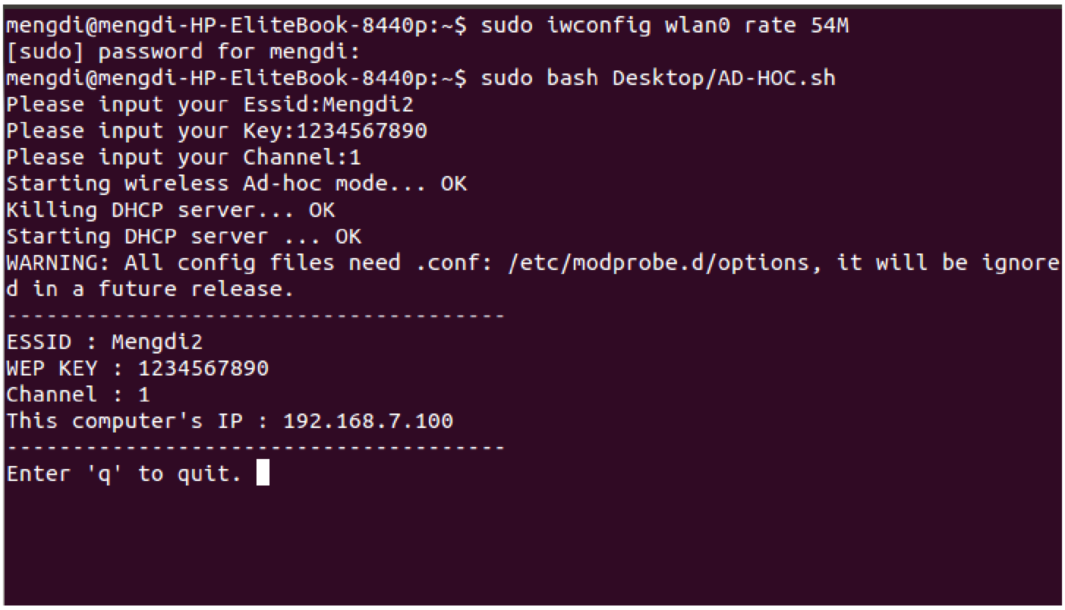
   Enable to send data packets from the server

   ```
   iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
   ```

   Enable the NAT table.
6. If the firewall rules are also enable on my server, more lines have to be specified to make sure that traffic from the wireless network is routed to the Internet and the firewall allows replies to that traffic to come back to the server.

   ```
   iptables -A FORWARD -i eth0 -o wlan0 -m state --state RE-
   LATED,ESTABLISHED -j ACCEPT
   iptables -A FORWARD -i wlan0 -o eth0 -j ACCEPT
   ```
7. Last the IP forwarding has to be enabled.

   ```
   sudo sh -c "echo 1 > /proc/sys/net/ipv4/ip_forward"
   ```



```
mengdi@mengdi-HP-EliteBook-8440p:~$ sudo iwconfig wlan0 rate 54M
[sudo] password for mengdi:
mengdi@mengdi-HP-EliteBook-8440p:~$ sudo bash Desktop/AD-HOC.sh
Please input your Essid:Mengdi2
Please input your Key:1234567890
Please input your Channel:1
Starting wireless Ad-hoc mode... OK
Killing DHCP server... OK
Starting DHCP server ... OK
WARNING: All config files need .conf: /etc/modprobe.d/options, it will be ignore
d in a future release.
--------------------------------------
ESSID : Mengdi2
WEP KEY : 1234567890
Channel : 1
This computer's IP : 192.168.7.100
--------------------------------------
Enter 'q' to quit.
```

Figure 11. Ad-Hoc Set up

The commands are written in the file “AD-HOC.sh” under the desk top. Then the shell script is run, the result is shown in Figure 11.

It will display the server’s ESSID, WEP key, IP address and channel number. According to the information, the client can find and connect to the server easily.

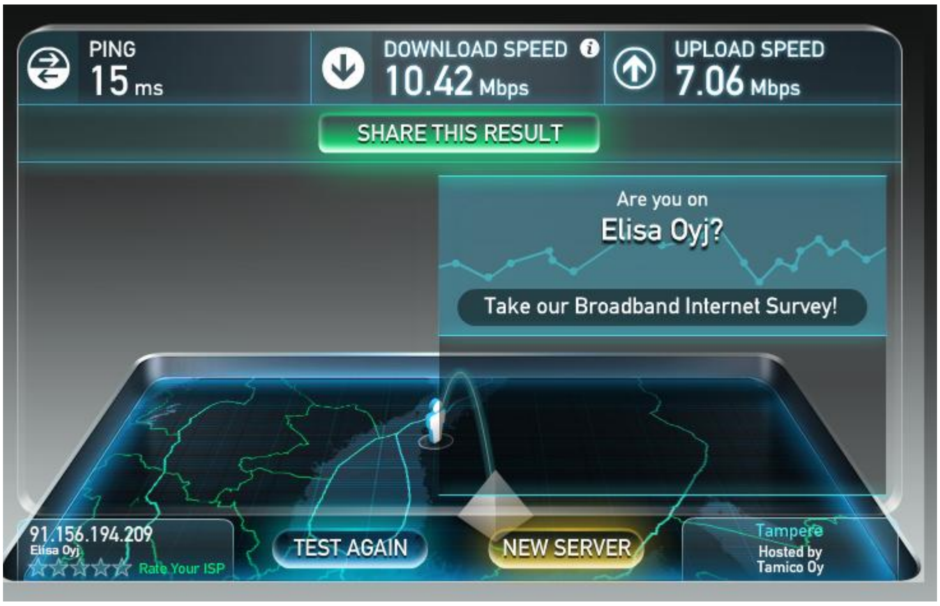


Figure 12. Client Speed Test

After the entire configuration, the client machine will be able to surf the Internet now. A speed test is done to check out the general download speed and upload speed, as shown in Figure 12.

3.2.3 Summary

For the establishment, now the client machine will be able to use the Internet through the Ad Hoc access point.

Compared to the Ad Hoc wireless network, the infrastructure mode has a better scale, centralized security management and improved reach. But in this case, the WLAN interface does not support the access point mode with hosapd, so the Infrastructure mode was discarded and Ad Hoc mode chosen instead to finish the job. Due to the situation, not all the mobile devices can connect to the access point. It puts up a limit to the entire work.

4 WIRELESS NETWORK PERFORMANCE TEST

In this section, Iperf the measure tool is used to test the performance of the network. By observing the jitter, bandwidth and packet loss, it is defined if the channel allocation choice is right and the performance of the wireless network has been improved.

Also this test should be arranged not only for one time, because it has to be done in a really crowded location e.g. the student accommodation. The more test, the more reliable result it is. The testing time also need to be in a day's different time. The system diagram shows in Figure 13.

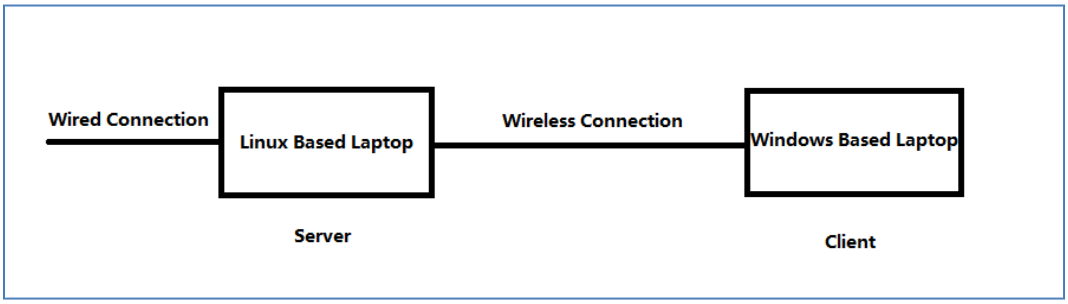


Figure 13. Iperf test diagram

4.1 Background

4.1.1 Iperf

Iperf is an open source C++ language program, which can measure network performance such as throughput, jitter and datagram loss. It is easy to get access through its own official page. It can be used on Windows, Linux, MacOS X and Oracle Solaris.

At least two machines are needed to do the test, one act as a server and the other acts as a client. In this case, the Linux machine acts as server and the windows machine acts as client(s). The server and the client can be easily set with two simple commands.

```
Server> iperf –s

Client> iperf -c “server IP address”
```

For the function part, it offers TCP bandwidth and UDP test. Iperf features can be seen as following: /8/

TCP:

1. Measure bandwidth.
2. Report MSS/MTU size and observed read size.
3. Support for TCP window size via socket buffers.
4. Client and server can have multiple simultaneous connections.

For UDP:

1. Client can create UDP streams with certain bandwidth.
2. Measure packet loss.
3. Measure delay jitter.
4. Multicast capable.

4.2 Configuration

TCP is used to measure the bandwidth and UDP to measure the packet loss, delay jitter and bandwidth to observe the performance of the network. The Linux based laptop will act as a server and the windows machine act as client(s).

First, a TCP connection is set. The following commands are done.

```
Server> iperf –s

Client> iperf -c “server IP address” –t 3600
```

Second, a UDP connection test is done. The commands are as follows:

```
Server> iperf –s –u –i 1

Client> iperf -c “server IP address” –t 3600 –u –i 1 –b 10m
```

4.3 Test

4.3.1 TCP Connection Test

The TCP connection helps to find out the WLAN’s maximum bandwidth. With this result, it can be compared if the internet performance has been improved or not. The Linux laptop will act as a server and Windows laptop act as a client. All the information will be collected on client side. The testing time is set to one hour to collect a stable result. It is shown in Figure 14 and Figure 15.

```
C:\Users\Mira>iperf -c 192.168.7.100 -t 3600 -i 1
------------------------------------------------------------
Client connecting to 192.168.7.100, TCP port 5001
TCP window size: 64.0 KByte (default)
------------------------------------------------------------
[  3] local 192.168.7.107 port 63778 connected with 192.168.7.100 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0- 1.0 sec  1.75 MBytes  14.7 Mbits/sec
[  3]  1.0- 2.0 sec  2.00 MBytes  16.8 Mbits/sec
[  3]  2.0- 3.0 sec  2.00 MBytes  16.8 Mbits/sec
[  3]  3.0- 4.0 sec  2.00 MBytes  16.8 Mbits/sec
[  3]  4.0- 5.0 sec  2.00 MBytes  16.8 Mbits/sec
[  3]  5.0- 6.0 sec  2.00 MBytes  16.8 Mbits/sec
[  3]  6.0- 7.0 sec  2.00 MBytes  16.8 Mbits/sec
[  3]  7.0- 8.0 sec  2.12 MBytes  17.8 Mbits/sec
[  3]  8.0- 9.0 sec  1.75 MBytes  14.7 Mbits/sec
[  3]  9.0-10.0 sec   640 KBytes  5.24 Mbits/sec
[  3] 10.0-11.0 sec   512 KBytes  4.19 Mbits/sec
```

Figure 14. Client TCP Connection

```
mengdi@mengdi-HP-EliteBook-8440p:~$ iperf -s
------------------------------------------------------------
Server listening on TCP port 5001
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[  4] local 192.168.7.100 port 5001 connected with 192.168.7.107 port 63630
[ ID] Interval       Transfer     Bandwidth
[  4]  0.0-3600.0 sec  6.34 GBytes  15.1 Mbits/sec
```

Figure 15. Server TCP Connection

4.3.2 UDP Connection Test

Now the UDP connection is started to see the Jitter and packet loss rate. These parameters help to find out how stable the network connection is. The Linux laptop will act as a server and Windows laptop act as a client. All the information from the server side is collected. It is shown in Figure 16 and Figure17.

```
C:\Users\Mira>iperf -c 192.168.7.100 -i 1 -t 3600 -u -b 10m
------------------------------------------------------------
Client connecting to 192.168.7.100, UDP port 5001
Sending 1470 byte datagrams
UDP buffer size: 64.0 KByte (default)
------------------------------------------------------------
[  3] local 192.168.7.107 port 54373 connected with 192.168.7.100 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0- 1.0 sec  1.19 MBytes  10.0 Mbits/sec
[  3]  1.0- 2.0 sec  1.19 MBytes  10.0 Mbits/sec
[  3]  2.0- 3.0 sec  1.19 MBytes  10.0 Mbits/sec
[  3]  3.0- 4.0 sec  1.19 MBytes  10.0 Mbits/sec
[  3]  4.0- 5.0 sec  1.19 MBytes  10.0 Mbits/sec
[  3]  5.0- 6.0 sec  1.19 MBytes  10.0 Mbits/sec
[  3]  6.0- 7.0 sec  1.19 MBytes  10.0 Mbits/sec
[  3]  7.0- 8.0 sec  1.19 MBytes  10.0 Mbits/sec
[  3]  8.0- 9.0 sec  1.19 MBytes  10.0 Mbits/sec
[  3]  9.0-10.0 sec  1.19 MBytes  10.0 Mbits/sec
[  3] 10.0-11.0 sec  1.19 MBytes  10.0 Mbits/sec
[  3] 11.0-12.0 sec  1.19 MBytes  10.0 Mbits/sec
[  3] 12.0-13.0 sec  1.19 MBytes  10.0 Mbits/sec
[  3] 13.0-14.0 sec  1.19 MBytes  10.0 Mbits/sec
```

```
[  3] 3591.0-3592.0 sec  1.19 MBytes  10.0 Mbits/sec
[  3] 3592.0-3593.0 sec  1.19 MBytes  10.0 Mbits/sec
[  3] 3593.0-3594.0 sec  1.19 MBytes  10.0 Mbits/sec
[  3] 3594.0-3595.0 sec  1.19 MBytes  10.0 Mbits/sec
[  3] 3595.0-3596.0 sec  1.19 MBytes  10.0 Mbits/sec
[  3] 3596.0-3597.0 sec  1.19 MBytes  10.0 Mbits/sec
[  3] 3597.0-3598.0 sec  1.19 MBytes  10.0 Mbits/sec
[  3] 3598.0-3599.0 sec  1.19 MBytes  10.0 Mbits/sec
[  3] 3599.0-3600.0 sec  1.19 MBytes  10.0 Mbits/sec
[  3]  0.0-3600.0 sec  4.19 GBytes  10.0 Mbits/sec
[  3] Sent 3061176 datagrams
[  3] Server Report:
[  3]  0.0-3600.5 sec  4.14 GBytes  9.87 Mbits/sec   0.745 ms 39569/3061175 (1.3
%)
[  3]  0.0-3600.5 sec  1 datagrams received out-of-order
```

Figure 16. Client UDP Connection

```
mengdi@mengdi-HP-EliteBook-8440p:~$ iperf -s -i 1 -u
------------------------------------------------------------
Server listening on UDP port 5001
Receiving 1470 byte datagrams
UDP buffer size:  208 KByte (default)
------------------------------------------------------------
[  3] local 192.168.7.100 port 5001 connected with 192.168.7.107 port 54373
[ ID] Interval       Transfer     Bandwidth        Jitter   Lost/Total Datagrams
[  3]  0.0- 1.0 sec  1.19 MBytes  9.96 Mbits/sec   0.567 ms    0/  847 (0%)
[  3]  1.0- 2.0 sec  1.20 MBytes  10.0 Mbits/sec   0.513 ms    0/  854 (0%)
[  3]  2.0- 3.0 sec  1.19 MBytes  9.97 Mbits/sec   0.766 ms    0/  848 (0%)
[  3]  3.0- 4.0 sec  1.19 MBytes  10.0 Mbits/sec   0.640 ms    0/  852 (0%)
[  3]  4.0- 5.0 sec  1.18 MBytes  9.93 Mbits/sec   0.688 ms    0/  844 (0%)
[  3]  5.0- 6.0 sec  1.20 MBytes  10.1 Mbits/sec   0.552 ms    0/  857 (0%)
[  3]  6.0- 7.0 sec  1.19 MBytes  10.0 Mbits/sec   0.696 ms    0/  850 (0%)
[  3]  7.0- 8.0 sec  1.19 MBytes  10.0 Mbits/sec   0.675 ms    0/  850 (0%)
[  3]  8.0- 9.0 sec  1.13 MBytes  9.46 Mbits/sec   0.781 ms    1/  805 (0.12%)
```

```
[  3] 3585.0-3586.0 sec   711 KBytes  5.82 Mbits/sec   1.006 ms    6/  501 (1.2%)
[  3] 3586.0-3587.0 sec   755 KBytes  6.19 Mbits/sec   0.826 ms    5/  531 (0.94%)
[  3] 3587.0-3588.0 sec   827 KBytes  6.77 Mbits/sec   0.909 ms    5/  581 (0.86%)
[  3] 3588.0-3589.0 sec  1.53 MBytes  12.8 Mbits/sec   0.632 ms    0/ 1089 (0%)
[  3] 3589.0-3590.0 sec  1.44 MBytes  12.1 Mbits/sec   0.962 ms    0/ 1025 (0%)
[  3] 3590.0-3591.0 sec  1.51 MBytes  12.6 Mbits/sec   0.725 ms    0/ 1075 (0%)
[  3] 3591.0-3592.0 sec  1.72 MBytes  14.5 Mbits/sec   0.806 ms    0/ 1229 (0%)
[  3] 3592.0-3593.0 sec  1.67 MBytes  14.0 Mbits/sec   0.664 ms    0/ 1194 (0%)
[  3] 3593.0-3594.0 sec  1.68 MBytes  14.1 Mbits/sec   0.749 ms    0/ 1196 (0%)
[  3] 3594.0-3595.0 sec  1.59 MBytes  13.3 Mbits/sec   0.680 ms    0/ 1134 (0%)
[  3] 3595.0-3596.0 sec  1.63 MBytes  13.7 Mbits/sec   0.746 ms    0/ 1162 (0%)
[  3] 3596.0-3597.0 sec  1.52 MBytes  12.7 Mbits/sec   0.880 ms    0/ 1081 (0%)
[  3] 3597.0-3598.0 sec  1.32 MBytes  11.1 Mbits/sec   0.626 ms    0/  944 (0%)
[  3] 3598.0-3599.0 sec  1.19 MBytes  9.98 Mbits/sec   0.301 ms    0/  849 (0%)
[  3] 3599.0-3600.0 sec  1.20 MBytes  10.1 Mbits/sec   0.663 ms    0/  855 (0%)
[  3]  0.0-3600.5 sec  4.14 GBytes  9.87 Mbits/sec   0.745 ms 39569/3061175 (1.3%)
[  3]  0.0-3600.5 sec  1 datagrams received out-of-order
```

Figure 17. Server UDP Connection

On the server side, the report is gathered every one second. It is easy to observe how well the network performance is. The testing time is set to one hour. In the end, Iperf will give us a final report. With the Jitter and packet loss rate, it is easy to find out which network is more stable.

4.4 Result Analyze

4.4.1 PSI Test Result

From the PSI calculation, all the channel information can be revived. In Figure 18 it can be seen, that channel 1 and 2 seems to be the most crowded channels and channel 8 is the least crowded channel.

```
Channel 1-> PSI=0.00000015344783114747mW
Channel 2-> PSI=0.00000015344783114747mW
Channel 3-> PSI=0.00000002880408660813mW
Channel 4-> PSI=0.00000002379221427185mW
Channel 5-> PSI=0.00000008334143945139mW
Channel 6-> PSI=0.00000005954922517954mW
Channel 7-> PSI=0.00000005954922517954mW
Channel 8-> PSI=0.00000000000000000000mW
Channel 9-> PSI=0.00000004747170852629mW
Channel 10-> PSI=0.00000009115330991200mW
Channel 11-> PSI=0.00000009115330991200mW
Channel 8 have better signal
mengdi@mengdi-HP-EliteBook-8440p:~$
```

Figure 18. PSI Result

4.4.2 TCP Connection Result

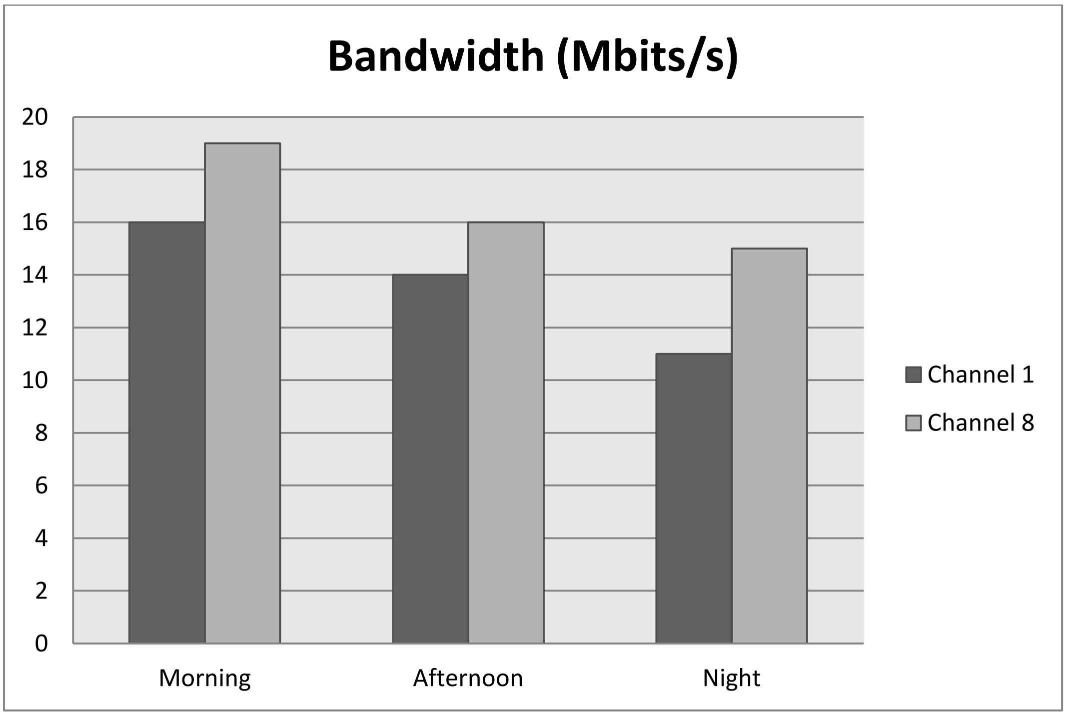


Figure 19. TCP Connection Result

Figure 19 shows that people usually use the Internet at night. There is almost a 4 Mbits/s bandwidth difference between morning and night. This is true e.g. with students. Students are not at home in the morning and day time, but at night they probably are keen Internet users. Moreover, the network on channel 1 is around 2-3 Mbits/s slower than channel 8.

All these facts show us that the network of the least crowded channel's network is faster than the most crowded channel. This result shows us there is at least improved internet speed no matter what time of the day it is.

4.4.3 UDP Connection Result

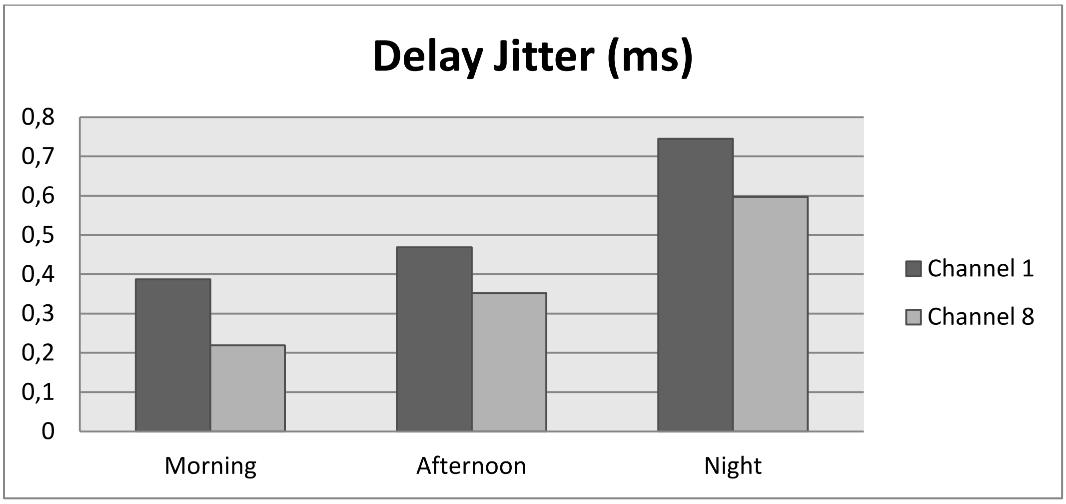


Figure 20. UDP Connection Delay Jitter result

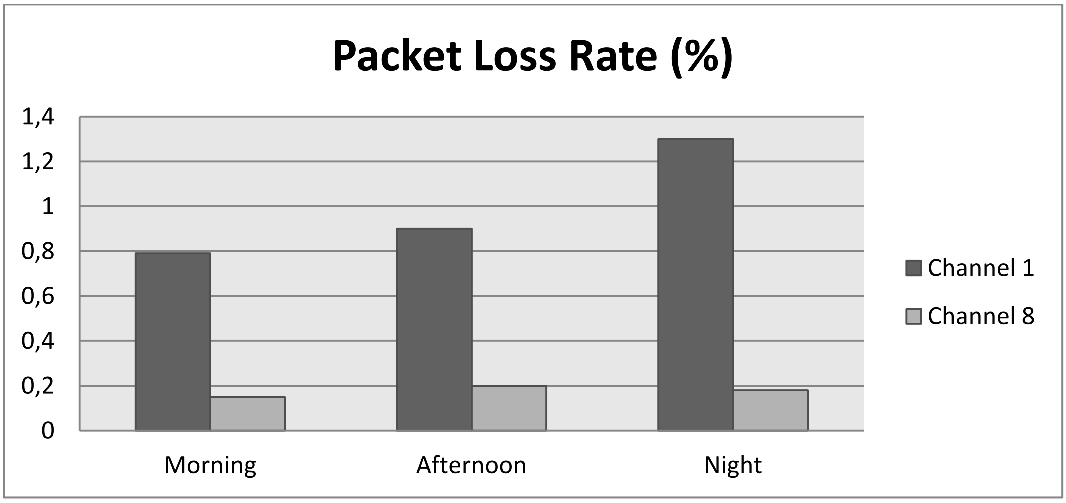


Figure 21. UDP Connection Packet Loss Rate result

From the UDP test, basically two important parameters are revived which are delay jitter and packet loss rate. With these two parameters, it can be seen how stable the wireless network is.

In Figures 20 and 21, it can be seen the jitter and packet loss rate are becoming higher and higher. This is because of the amount of the other network users will interference our own network.

But from two different channels results, channel 8 is much less than channel 1. When the test is at night, the packet loss rate in the crowded channel can be more than 80% compared to the better channel. The high jitter and packet loss rate may cause the lag when you play a game or do the video chat.

4.5 Summary

In this section, the TCP and UDP test were run a few times not only at the same time of the day. According to the real situation, people are not using internet all the time. So the test was completed at different times of the day to check out how the channel allocation method works when it faces with different situations. The bigger bandwidth, lower jitter and packet loss rate shows our method has improved the performance of wireless network.

5 CONCLUSION

In this project it was proved that the WLAN channel allocation can affect the performance of the wireless connection directly.

By using the channel allocation method, the better channel can be found in one area.

Changing the default channel number into another less crowded one can directly affect the performance quality of our own wireless network. Also, the improvement of our own network can also help the wireless environment for other wireless network user in one area.

By reducing the delay jitter and packet loss rate, the wireless network will be more stable and get greatly improved.

During this project, I have learned a lot about WLAN. This is directly related to our daily life. However, we can still have some WLAN interferences by other things. At least we find one way to improve it greatly.

REFERENCES

/1/ WLAN diagram. Accessed 10.02.2014 http://arxiv.org/ftp/arxiv/papers/1307/1307.2661.pdf

/2/ "IEEE 802.11-2012: 16.4.6 - PMD Operating Specifications, General". IEEE. 2013-05-15.

/3/ 2.4 GHz Band Channels. Accessed 10.02.2014 http://upload.wikimedia.org/wikipedia/commons/8/8c/2.4_GHz_WiFi_channels_%28802.11b%2Cg_WLAN%29.svg

/4/ IEEE Standard for Information Technology. Telecommunications and Information Exchange between Systems. Local and Metropolitan Area Networks-specific Requirements. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. IEEE STD 802.11-2007 (Revision of IEEE Std 802.11-1999)

/5/ Interference. Accessed 20.03.2014 http://www.rm.com/support/technicalarticle.asp?cref=tec597527

/6/ Spectral Masks for 802.11g in the 2.4 GHz Band. Accessed 03.15.2014 http://upload.wikimedia.org/wikipedia/commons/a/ac/2.4_GHz_WiFi_channels_%28802.11g_WLAN%29.svg

/7/ Sauter Martin (2010). "3.7.1 Mobility Management in the Cell-DCH State". From GSM to LTE: An Introduction to Mobile Networks and Mobile Broadband (eBook). John Wiley & Sons. p. 160.

/8/ Iperf features. Accessed 20.03.2014 http://iperf.fr/

APPENDIX 1. PSI Calculation

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
int main(){
	FILE *fp;
	int chary[50], tempch, i, length,charypro[50];
	double RSSI[50], temprssi,RSSIpro[50];


	int chlength=0;
	system("iwlist   wlan0   scan|grep   -o   -e   'Channel:\\([0-9]\\+\\)'   -e   'Signal   level=\\([-0-9]\\+\\)'|   perl   -e 'for(<>){s/^.*://;s/^.*=//;print}'> result.txt");
	fp = fopen("result.txt", "r");
	if(fp==NULL){
		printf("File not available\n");
	}
	else
	{
		i = 0;
	     while( fscanf(fp, "%d%lf", &tempch, &temprssi) != EOF )
		{
			chary[i] = tempch;
			RSSI[i] = temprssi;
			i++;
		}
		printf("Read %d pieces of data\n", i);
		length = i;
		for(i=0; i<length; i++)
		{
			if(chary[i]<=11)
				{
					charypro[chlength]=chary[i];
					RSSIpro[chlength]=RSSI[i];
					chlength ++;
					printf("Channel    %d        Signal level%.2fdBm\n", chary[i], RSSI[i]);
				}
		}
		fclose(fp);
	}
	double Power[50],PSI[50];

	int channelNum=1,channelLastNum=1;

	printf("\n\n\nLength%d\n", chlength);

	for(i=0;i<chlength;i++)
	{
		PSI[i]=0.0;
		Power[i]=pow(10.0,RSSIpro[i]/10.0);
//
	for(channelNum=1;channelNum<=chary[chlength];channelNum++)

		printf("Channel    %d    ->    %.2fdBm->    %.20fmW\n", charypro[i],RSSIpro[i],Power[i]);
```

```
            PSI[charypro[i]]+=Power[i];

      }


      for(i=1;i<=11;i++)
      {
      //    printf("%d->%.15fmW\n", i,PSI[i]);
      }

double PSIsum[i];
      for(i=1;i<=11;i++)
      {
            PSIsum[i]=PSI[i-1]+PSI[i]+PSI[i+1];

            printf("Channel %d-> PSI=%.20fmW\n", i,PSIsum[i]);
      }
double x=PSIsum[0];
int best;
      for(i=1;i<=11;i++)
      {

    if (PSIsum[i]<x) {x=PSIsum[i];best=i;}

      }
      for(i=1;i<=11;i++)
      {
            if (PSIsum[i]==x)

      printf("Channel %d have better signal\n",i);
      }
}
```

APPENDIX 2. Ad Hoc Access Point Establishment

```
#!/bin/bash
#
# Wireless Ad-hoc script
# User variables
mywlan="wlan0"
myip="192.168.7.100"
mydhcprange="192.168.7.101,192.168.7.110"

# Main program
read -p "Please input your Essid:" myessid
read -p "Please input your Key:" mywepkey
read -p "Please input your Channel:" mychan

echo -n "Starting wireless Ad-hoc mode... "

ifconfig $mywlan down || exit 1
iwconfig $mywlan mode ad-hoc || exit 1
iwconfig $mywlan essid $myessid
[ "$mychan" ] &&iwconfig $mywlan channel $mychan
[ "$mywepkey" ] && iwconfig $mywlan key $mywepkey

ifconfig $mywlan $myip
ifconfig $mywlan up && echo "OK"
echo -n "Killing DHCP server... "
killall dnsmasq && echo "OK"
echo -n "Starting DHCP server ... "
dnsmasq --dhcp-range="$mydhcprange" && echo "OK"

iptables -F
iptables -P INPUT ACCEPT
iptables -P FORWARD ACCEPT
iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
iptables  -A  FORWARD  -i  eth0  -o  wlan0  -m  state  --state  RE-
LATED,ESTABLISHED -j ACCEPT
iptables -A FORWARD -i wlan0 -o eth0 -j ACCEPT
sudo sh -c "echo 1 > /proc/sys/net/ipv4/ip_forward"

echo "----------------------------------------"
echo "ESSID : $myessid"
[ "$mywepkey" ] && echo "WEP KEY : $mywepkey"
[ "$mychan" ] && echo "Channel : $mychan"
echo "This computer's IP : $myip"
echo "----------------------------------------"
while true; do
echo -n "Enter 'q' to quit. "
read value
if [ "$value" == "q" ]; then
break
fi
done

echo -n "Killing DHCP server... "
killall dnsmasq && echo "OK"
echo -n "Killing wireless... "

ifconfig $mywlan down
```

```
iwconfig $mywlan mode managed
iwconfig $mywlan essid off
iwconfig $mywlan key off
echo "OK"
echo "Wireless Ad-hoc mode terminated."
echo "Restarting your network manager....."
sudo service network-manager restart && echo "....OK"

exit 0
```